Electrical Modeling and Simulation for Stockpile Stewardship

A survey of radiation modeling and circuit simulation approaches that are essential for stockpile stewardship.

By Heidi K. Thornquist, Eric R. Keiter, and Sivasankaran Rajamanickam

DOI: 10.1145/2425676.2425685

With the elimination of underground nuclear testing, science-based stockpile stewardship requires increased reliance on high-performance modeling and simulation of weapon systems. Established in 1995, the Advanced Simulation and Computing (ASC) Program supports the U.S. Defense Programs' shift in emphasis from test-based confidence to simulation-based confidence. Under ASC, computer simulation capabilities are developed to analyze and predict the performance, safety, and reliability of nuclear weapons and to certify their functionality. These simulations are central to U.S. national security as they provide a computational surrogate for nuclear testing [1].

The ASC Program aims to model a large variety of physical and engineering phenomena and incorporate these models into integrated design codes. This includes, but is by no means limited to, the modeling of radiation, electrical, and electromagnetic effects. Since electrical system components are major elements in today's weapon systems, it is necessary to predict the reliability and survivability of weapon systems and components when exposed to hostile radiation environments and electromagnetic insults. This article will present select electrical modeling and simulation capabilities developed under the ASC Program, specifically those deployed through the Xyce project [2]. A basic introduction to electrical simulation will be followed by a discussion of radiation modeling and algorithms research for parallel simulation of modern stockpile technologies.

ELECTRICAL SIMULATION

Electrical simulation is a field that includes both device and circuit simulation techniques, each of which serves a distinct purpose. Device simulation is a higher fidelity approach, in which a single semiconductor device is represented with a set of coupled partial differential equations (PDEs), discretized on a spatial mesh. Device simulation is intended to be accurate, using models for the behavior of the electrical devices that are based on fundamental physics. However, device simulation is often compute-intensive and is not practical for simulation of entire circuits. Thus, transistor-level models (compact models) are derived from these physics-based simulations that are based on the underlying physics, empirical data (curve-fitting), or tabular data (look-up table). Compact models are much faster to simulate than the original physics-based model and can be integrated into a circuit simulator.

Circuit simulation is a technique for checking and verifying the design of electrical and electronic circuits and systems prior to manufacture and deployment [3]. This technique can be broken down into analog or digital (event-driven) approaches, as well as mixed-signal and mixed-mode simulation. Analog circuit simulation uses a detailed, transistor-level description of the circuit to generate a system of network-coupled differential algebraic equations. For integrated circuit

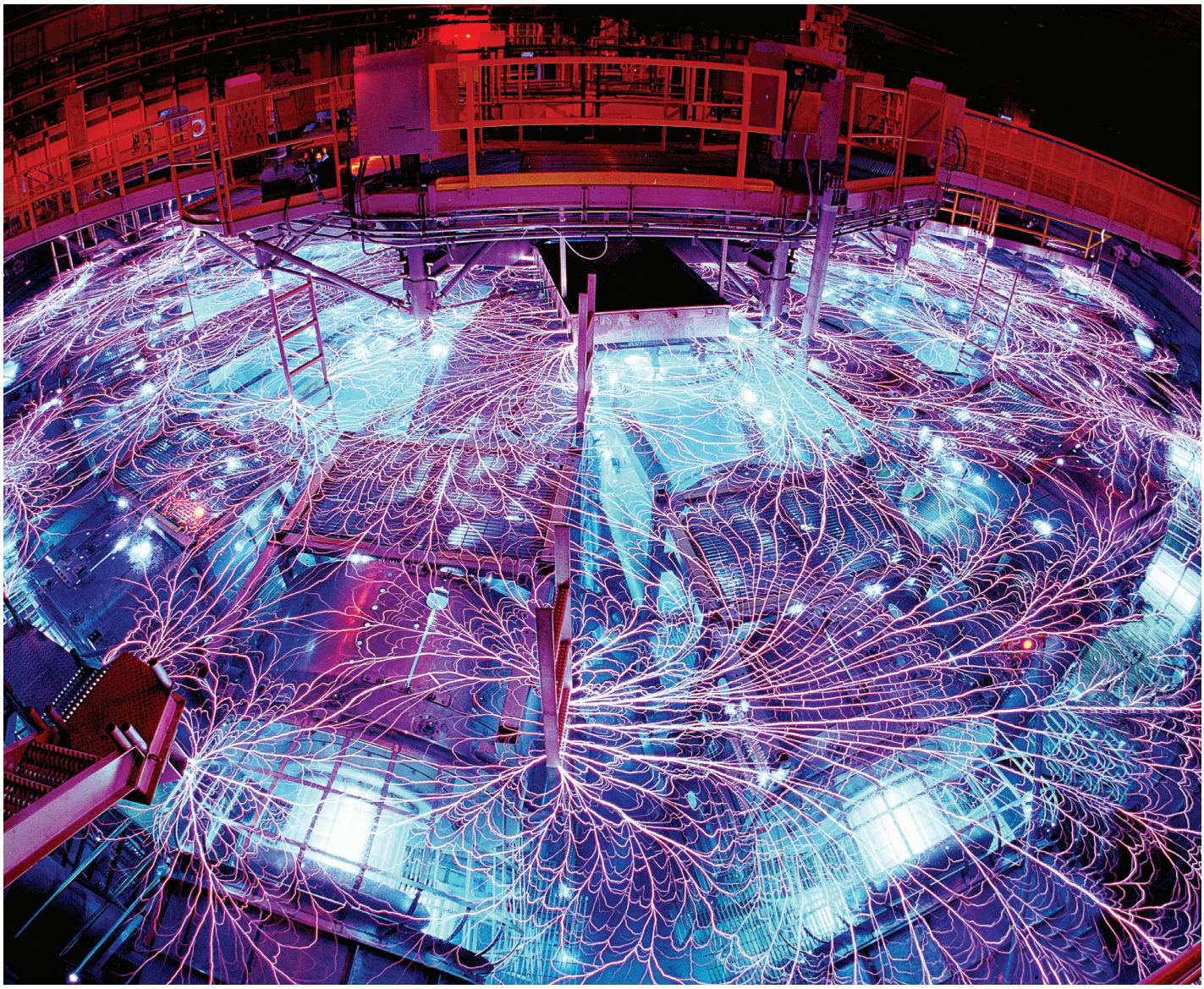


Image courtesy of Sandia National Laboratories.

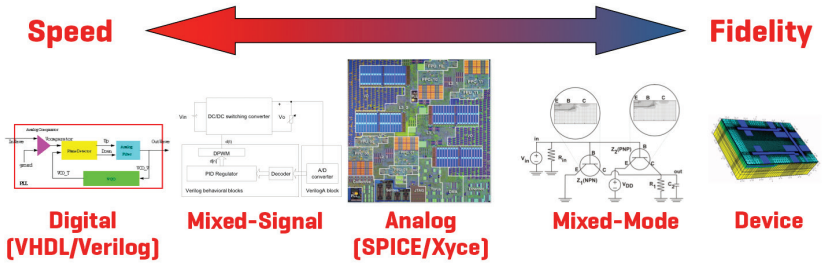
design, time-domain (transient) analog circuit simulation is an essential, yet expensive, part of the computer-aided design process. Digital circuit simulation is used predominately for verifying timing characteristics like propagation time and rise/fall time delays for circuits containing only digital components. Mixed-signal simulation can greatly reduce the analog simulation time by replacing analog models with event-driven models for digital devices. Conversely, mixed-mode simulation adds fidelity by replacing analog models with PDE-based devices.

In the design flow process, the entire continuum of device and circuit simulation approaches are used. The process leverages the tradeoff between speed and fidelity to perform device verification (see Figure 1). However, when new semiconductor device technologies are designed, a compact model has to be produced as well to enable faster analog and/or mixed-signal simulation.

COMPACT MODELING OF RADIATION EFFECTS

Compact modeling is the foundational model for circuit simulation. Most electrical compact models are designed to provide a mathematical current-voltage (I-V) relationship to a circuit simulator. The circuit simulator will simulate a network of such devices, and use these I-V relationships to enforce Kirchhoff's laws across the network. Ideally, a compact model would be developed from a mathematically precise solution to the device equations. However, for most devices of interest, the set of equations is nonlinear, multidimensional, and com-

Figure 1. Continuum of device and circuit simulation approaches.



plex, so an exact mathematical solution is not feasible.

Instead, compact models are derived from a combination of physical intuition, simplifying assumptions, empirical approximations, and (finally, after many simplifications have been applied) mathematical solutions. This combination can be hard to achieve in practice, but over the years a set of common approximations for semiconductor devices have been established.

The simulation of ionizing radiation effects in transistors and circuits is one of the purposes for which Xyce was designed. Compact models of current due to radiation-induced ionization, also known as "photocurrent," have been developed to support this effort. Photocurrent in semiconductor devices (such as diodes and transistors) is created when ionizing radiation imparts enough energy on valence band electrons for them to be exited into the conduction band of the device, and thus become mobile. When this happens, these newly mobile electrons leave behind positively charged holes in the valence band, which also become mobile and contribute to photocurrent. Under ionizing conditions, these newly excited electrons and holes are considered to be "excess carriers." Excess carriers result in additional currents not present under normal circuit operating conditions. At high enough magnitudes the circuit could temporarily be unpowered.

To model this effect in a compact model, it is important to distill the transport behavior to as small a set of equations as possible. Fortunately, excess carriers can be treated separately from the other carriers in the device. Also, despite the fact that electrons and holes are oppositely charged, they can be treated as a single species, as long as the carrier densities are high enough for them to establish internal electric fields coupling them together. This is important because it means that photocurrent can be modeled with a single transport equation, known as the ambipolar diffusion equation (ADE), which Van Roosbroeck [4] first suggested for semiconductors in 1953:

$$\frac{\partial u}{\partial t} = D_a \nabla^2 u - \mu_a E \cdot \nabla u - \frac{u}{\tau} + g(t)$$

where u is the concentration of the excess carrier, E is the electric field, τ is the carrier lifetime, $g(t)$ is the source rate due to radiation, D_a is the ambipolar diffusion constant, and μ_a is the ambipolar mobility. Photocurrent is determined from the following expression, evaluated at the boundary:

$$J(t) = qD_a \frac{\partial u(x,t)}{\partial x}$$

So, if a solution to the ADE can be found, current can be estimated. The ADE has a number of published analytic solutions, which usually assume a one-dimensional geometry. Most also make other assumptions, such as an infinite domain. The most complete one-dimensional solution was published by Axness et al. [5], in 2004, and is based on using finite Fourier sine transforms to transform the ADE into the nonhomogeneous heat equation.

$$u(x,t) = V(x,t)\, e^{-a \cdot x + bt}$$

$$\frac{\partial V}{\partial t} = D_p \nabla^2 V + g(t)\, e^{-a \cdot x + bt}$$

The finite Fourier transform technique eventually yields the following expression for current in a highly doped diode:

$$J(t) = 4q\left[\frac{L^2}{w}\right] \sum_{n=0}^{\infty} \int_0^{t_p} g(t - u\tau)\, e^{-a_{2n+1} u}\, du$$

Where L is the diffusion length and w is the width of the neutral region. An equation like this is applied to each neutral region of a semiconductor device model to obtain the total ionization current for that device. By incorporating expressions like this in existing transistor models, the additional behavior due to ionizing radiation is modeled. When applied to large circuits, a simulator like Xyce is able to simulate collective photocurrent effects, such as power rail collapse.

It is necessary to predict the reliability and survivability of weapon systems and components when exposed to hostile radiation environments and electromagnetic insults.

PARALLEL CIRCUIT SIMULATION

Efficient, scalable circuit simulation is important for modern stockpile technologies, where parasitic effects can increase the device count in an integrated circuit by an order of magnitude or more. Traditional transistor-level simulation, originally made popular by the Berkeley SPICE program [6], becomes impractical beyond tens of thousands of devices due to the reliance on sparse direct linear solvers. Many attempts have been made to allow for faster, large-scale, transistor-level circuit simulation. Fast-SPICE tools use event-driven simulation techniques and lookup tables for pre-computed analog device evaluations. Additionally, they often use circuit-level partitioning algorithms and more efficient data structures to enable the simulation of much larger problems [7, 8]. Unfortunately, the approximations inherent to these simulation approaches can break down under some circumstances, rendering such tools unreliable.

With the transition from single-core processors to multicore processors, parallel transistor-level simulation has received more interest from the electronic design automation community. Developed through the ASC Program, Xyce is a parallel transistor-level simulator whose purpose is to support the simulation of radiation effects on electronics. For SPICE-style simulators, like Xyce, the circuit is described by a netlist file, which lists the individual components and how they are connected together. This list of devices and interconnectivity is transformed via modified nodal analysis into a set of nonlinear differential algebraic equations (DAEs):

$$\frac{dq(x(t))}{dt} + f(x(t)) = b(t),$$

where $x(t) \in \Re^N$ is the vector of circuit unknowns, q and f are functions representing the dynamic and static circuit elements (respectively), and $b(t) \in \Re^M$ is

the input vector.

Time-domain (transient), transistor-level simulation implicitly solves the set of nonlinear DAEs above through numerical integration methods, resulting in a nested solver loop. Numerical integration methods require the solution to a sequence of nonlinear equations, $F(x) = 0$. Typically, Newton's method is used to solve these nonlinear equations, which generates a sequence of linear systems, $Ax = b$, that involve the conductance,

$$G(t) = \frac{df}{dx}(x(t)),$$

and capacitance,

$$C(t) = \frac{dq}{dx}(x(t)),$$

matrices. For DC (steady state) analysis, which is often used to provide an initial guess for transient analysis, the q-terms are not present, so the linear system only involves the conductance matrix.

The computation time in transistor-level circuit simulation is dominated by repeatedly solving the linear system of equations, which is at the center of the nested solver loop. This requires the assembly of the linear system, which involves device evaluations for the whole circuit followed by the insertion of device contributions into the Jacobian matrix and residual vector. So, the most dominant part of the simulation time can be broken down into the device loads (device evaluation plus matrix and vector assembly) and the numerical method used to solve the linear system. For smaller problems, the time needed to perform the device loads dictates the total simulation time. However, as the size of the circuit increases, the linear solution method will start to dominate the computation time.

Since this article is addressing modern stockpile technologies, the focus will be on scalable linear solution methods for circuit simulation.

SCALABLE LINEAR SOLUTION METHODS

The Jacobian matrices generated in circuit simulation are typically sparse, have heterogeneous non-symmetric structure, and are often ill-conditioned.

Figure 2. Graph/Hypergraph based on unsymmetric (left) and symmetric (right) ordering of the sparse linear system for parallelism in ShyLU.

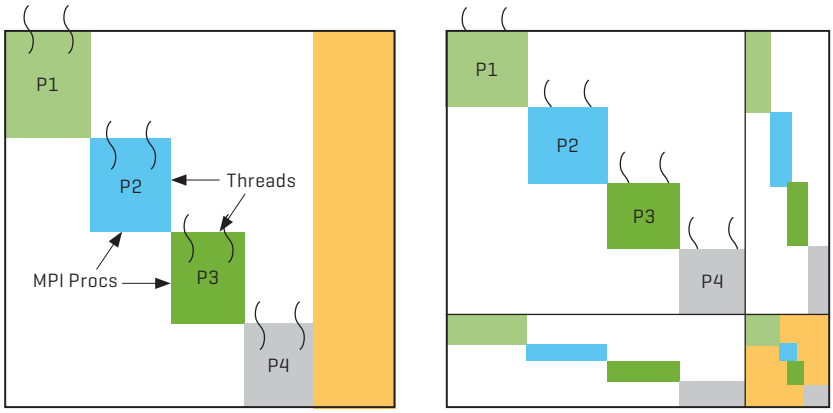
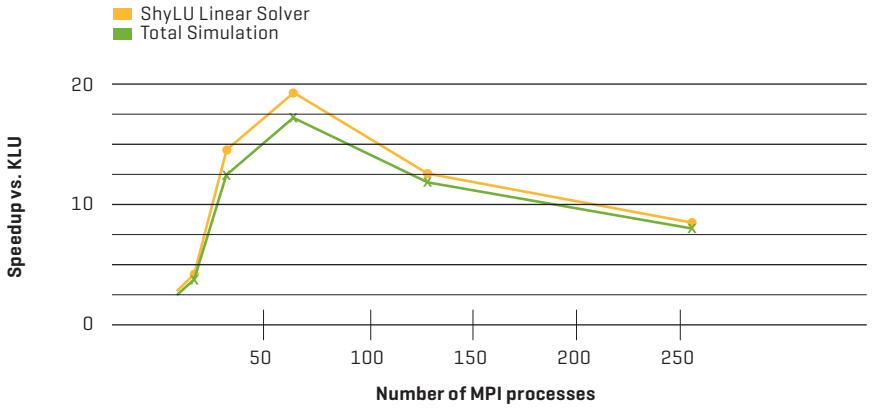


Figure 3. Strong scaling of Xyce simulation time and ShyLU linear solver time for MPI only execution of application-specific integrated circuit design.



Direct sparse linear solvers [9, 10] are the industry standard approach because of their reliability and ease of use. This is understandable for smaller linear systems, because direct solvers are usually faster than their iterative counterparts. However, when the linear system has hundreds of thousands of unknowns or more, direct solvers become less practical as they suffer from poor scaling.

Despite the problems inherent to circuit matrices, iterative solvers have the potential to be a scalable solution method for large-scale linear systems with lower algorithmic complexity. They are not as easy to use as direct solvers because their effectiveness is dependent upon finding an adequate preconditioner. However, progress has been made on the use of parallel iterative methods in transient analysis. Unfortunately, many types of preconditioning techniques suffer from the fact that the number of iterations to solve the linear system will increase with the number of message passing interface (MPI) processes.

Recent developments in linear solution methods for large-scale circuit simulation have focused on Schur complement techniques to bridge the gap between direct and iterative methods, utilizing both to create a solver that is scalable and robust. While these new techniques have been integrated into Xyce, they are being developed in the ShyLU package within Trilinos [11]. ShyLU is a hybrid linear solver in both

the parallel programming sense—using MPI and threads—and in the mathematical sense, using features from direct and iterative methods [12].

The Schur complement decomposition of a linear system is as follows. Let $Ax = b$ be the linear system of interest. Suppose A has the form:

$$A = \begin{pmatrix} D & C \\ R & G \end{pmatrix}$$

where D and G are square and D is nonsingular. The Schur complement after eliminating the top row is $S = G - R*D^{-1}) * C$. Solving $Ax = b$ then consists of solving

$$\begin{pmatrix} D & C \\ R & G \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

through these steps

1. $D_z = b_1$.
2. $Sx_2 = b_2 - Rz$.
3. $Dx_1 = b_1 - Cx_2$.

ShyLU uses graph/hypergraph partitioning to permute the matrix into the form above, where D is a block diagonal matrix. This form, illustrated in Figure 2, is called bordered block diagonal form. In general, each diagonal block in D corresponds to an MPI rank and is factored using a direct solver. This allows for steps 1 and 3 to be scalable. In step 2, the Schur complement, S, is never formed explicitly. Instead, an approximation of S is used to precondition the iterative linear solver that will perform step 2.

The Schur complement approach developed in ShyLU enabled Xyce to simulate a large modern integrated circuit design for stockpile stewardship. The circuit design, including parasitic elements, has more than 2 million devices and generates a linear system with around 1.9 million unknowns. A transient simulation of this circuit was performed on up to 256 MPI processes of a commodity cluster. The results shown in Figure 3 indicate a 20x speedup in the linear solve time when compared to the KLU direct solver [9]. Figure 3 also illustrates the performance for this approach is optimal for 64 MPI processes and deteriorates for 128 and 256 MPI processes. This is caused by an imbalance in the matrix partitioning with 128 or more MPI processes, such that some MPI processes have no rows in the bordered block diagonal form (see Figure 2). While these results are obtained using only MPI, threads may easily be employed in this framework by using a multithreaded direct solver for D or by multi-threading the multiple right hand side triangular solves, D^(-1)*C, for the Schur complement. Adding this fine-grained parallelism to ShyLU will enable further speedups in the linear solve and total simulation time.

With the transition from single-core processors to multicore processors, parallel transistor-level simulation has received more interest from the electronic design automation community.

CONCLUSION

Radiation modeling and circuit simulation are essential for stockpile stewardship. The development of more accurate models for ionizing radiation effects in transistors and circuits will help researchers study collective photocurrent effects, such as power rail collapse, while performing fewer tests on current and next-generation electronics. Developing parallel, scalable solution methods for the large linear systems of equations generated through the simulation of modern stockpile technologies will help researchers validate design decisions. Both areas of research serve the nuclear stockpile by addressing the need for simulation-based confidence mechanisms.

References

[1] Advanced Simulation and Computing (ASC) Program Website. http://nnsa.energy.gov/asc

[2] Xyce Parallel Electronic Simulator Website. http://xyce.sandia.gov

[3] Najm, F. N. *Circuit Simulation*. John Wiley & Sons, Inc., 2010.

[4] Van Roosbroeck, W. The transport of added current carriers in a homogeneous semiconductor. *Physical Review* 91, 2 (1953), 282–289.

[5] Axness, C. L., et al. Analytic light-or radiation-induced pn junction photocurrent solutions to the multidimensional ambipolar diffusion equation. *Journal of Applied Physics* 96, 5 (2004), 2646–2655.

[6] Nagel, L. W. SPICE2, a computer program to simulate semiconductor circuits. University of California, Berkeley. Technical report. Memorandum ERL-M250, 1975.

[7] Newton, A. R., and Sangiovanni-Vincentelli, A. L. Relaxation based electrical simulation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* CAD-3, 4 (1984), 308–330.

[8] White, J. K., and Sangiovanni-Vincentelli, A. *Relaxation Techniques for the Simulation of VLSI Circuits*. Kluwer Academic Publishers, 1987.

[9] Davis, T. A. *Direct Methods for Sparse Linear Systems*. SIAM, 2006.

[10] Kundert, K. Sparse matrix techniques. In A. Ruehli (ed.), *Circuit Analysis, Simulation and Design*. North-Holland, 1987.

[11] Heroux, M. A., et al. An overview of the Trilinos project. *ACM Transactions on Mathematical Software* 31,3 (2005), 397–423.

[12] Rajamanickam, S., Boman, E., and Heroux, M. ShyLU: A hybrid-hybrid solver for multicore platforms. In *Proceedings of the 26th International Parallel Distributed Processing Symposium* (Shanghai, May 21-25). IEEE, Washington, D.C., 2012, 631 –643.

Biographies

Heidi K. Thornquist received her B.A. degree in mathematics from Humboldt State University in Arcata, CA in 1998. She received M.A. and Ph.D. degrees in computational and applied mathematics from Rice University in 2003 and 2006, respectively. In 2003, she joined Sandia National Laboratories of Albuquerque, NM, and is currently a Senior Member of Technical Staff. She joined the Trilinos Project in 2003 and was a co-recipient of the 2004 R&D 100 Award. In 2006, she joined the Xyce Project and was a co-recipient of the 2008 R&D 100 Award for the Xyce circuit simulator.

Eric R. Keiter received B.A. degrees in physics and mathematics from Augustana College in Rock Island, Illinois in 1989. He received M.S. and Ph.D. degrees in materials science from the University of Wisconsin in 1991 and 1996, respectively. From 1996 to 1998, he held a postdoctoral position at the University of Illinois in the electrical engineering department, where he did research on simulation of dusty plasmas. He joined Sandia National Laboratories of Albuquerque, NM in 1998, and is the principal investigator for the Xyce circuit simulator project. Keiter was a co-recipient of the 2008 R&D 100 Award for the Xyce circuit simulator.

Sivasankaran Rajamanickam received his B.E. in computer science and engineering from the Madurai Kamaraj University (India) in 1999 and his Ph.D. in computer engineering from the University of Florida in 2009. He joined Sandia National Laboratories in 2010 as a postdoctoral researcher is currently a staff member in the Electrical Systems Modeling department. Rajamanickam has worked on combinatorial algorithms and preconditioners for manycore architectures in Trilinos and Xyce libraries. He was also part of development teams at Sun Microsystems and Tata Consultancy Services.